PROGRAMS FOR CALCULATING RAILROAD EXCAVATION AND EMBANKMENT USING VISUAL BASIC 6
(draft)

by

James C. Burke

Introduction

The purpose of this study is to develop a package of useful programs using Microsoft Visual Basic 6 for calculating the volume of excavation or embankment for a railroad survey. The same applications can be applied to some aspects of estimating earthworks for highway construction. Some the programs contained in the package are devoted to profile leveling.

During the course of surveying a railroad cross-section calculations are made at each survey station. Short distances between stations allow for estimation of volume that has greater accuracy, but more time is committed to making calculation. The object of developing a computer program to handle the tedious formulas associated with this task is to facilitate the preparation of estimates in the field whereby the surveyor enters raw measurements. The program provides instance cross-sectional areas, the volume between stations, and keeps a running total. By utilizing the Visual Basic code included herein, all who are familiar with the software can quickly prepare the program for practical application, and avoid the expense of purchasing specialized software.

The Formulas

The basic formula for the cross-section of a railroad excavations and embankment, as expressed by John Woodbridge Davis in his *Formulae for the Calculation of Railroad Excavation and Embankment* of 1877, is the sum of the triangles formed by the division of the cut from the center of the rail bed. The height of the line from the center to the top of the cut is represented as c, the term b represents one half of the width of the road-bed, r is the elevation of the right slope stake above grade, l is the elevation of the left slope stake above grade, l' is the horizontal distance from the side left, r' is the horizontal distance from the side right, and w is the entire width between slope stakes. The entire formula can be expressed as

$$\frac{1}{2}rb + \frac{1}{2}lb + \frac{1}{2}c(r' + b) + \frac{1}{2}c(l' + b)$$

The ratio of the slope is denoted as *S*, where $S = \frac{r}{r'} = \frac{l}{l'}$ and $r = Sr', l = Sl'$ making the area of the section the expression $\frac{1}{2}Sb(r' + l') + \frac{1}{2}c(r' + l' + 2b)$, thus $\frac{1}{2}Sb(r' + l' + 2b) - Sb^2 + \frac{1}{2}c(r' + l' + 2b)$ or $\frac{1}{2}w(c + Sb) - Sb^2$. Davis concentrates on two variables for his formulas to determine volume, the center *c* and the width *w*. The volume of excavation between two stations is $(wc + w'c' + Sb(w + w') - 4Sb^2)\frac{D}{108}$ where *D* is the distance between stations and the results are given in cubic yards.

In the recent times, surveying text books such as *Surveying Theory and Practice* by Raymond Davis, Francis Foote, and Joe Kelly presented the formulas for excavations and embankments in a several simple forms for special applications rather than general formulas. The special applications are Regular Cross-sections, Three-level Sections, and Irregular Road Cross-sections. The Regular Cross-section method is the simplest, $A = c(d + \frac{w}{2})$. However, it was limited applications because the excavation or embankment is made on level ground. The most useful of the formulas is the Three-level Section, $A = \frac{w}{4}(h_1 + h_2) + \frac{c}{2}(d_1 + d_2)$. Like the formula presented by J.W. Davis, it allows for different depths or heights on the center, left, and right sides. The volume of earthworks is general calculated using in the Average End Areas formula, $V = \frac{l}{2}(A_1 + A_2)$ where the volume between the two stations is the average of their cross-sectional areas multiplied by the distance separating them. The volume between two cross-sections is the practical unit. The surveyor can then provide the volume of excavation or embankment at a particular location then calculates a total of the volumes for an estimate in the final report. These are the basic formulas that need to be utilized in a computer program.

The Irregular Road Cross-sections method is an extremely complex set of operations that are applied to cross-sectional areas that have ridges and depressions on the surface. The pattern is notated as the matrix $\left|\begin{matrix} H_2 & H_1 & C & h_1 & h_2 \\ D_2 & D_1 & 0 & d_1 & d_2 \end{matrix}\right|$ where the center line *C* set at 0 with the lower elements on the left of *C* negative and the lowers elements right of *C* positive and half the roadbed $(-\frac{w}{2}, +\frac{w}{2})$ is added to the opposite ends of the matrix. The final form of the matrix appears as $\left|\begin{matrix} 0 & H_2 & H_1 & C & h_1 & h_2 & 0 \\ -w/2+ & -D_2+ & -D_1+ & 0 & +d_1- & +d_2- & +w/2- \end{matrix}\right|$ and can be expressed in the operations that multiple the upper term by the sum of two adjacent lower terms, $H_2\left(+\frac{w}{2} - D_1\right) + H_1(+D_2 - 0) + C(+D_1 + d_1) + h_1(0 + d_2) + h_2(-d_1 + \frac{w}{2})$. There can be more divisions of the cross-section, for example $H_1,\ H_2,\ H_3 \ldots n$ and $d_1,\ d_2,\ d_3 \ldots n$. This type of method requires

different treatment in a software program as each cross-section could have different divisions of the surface. The result must be divided in half.

Project 1: Surveyor's Railroad Embankment and Excavation Calculator

This *Visual Basic 6* application is designed to assist the surveyor in the field with tools for determining the Regular Cross-Section, Three-level Section, and Volumes between stations. The program also features a log for note such as estimated volumes between particular stations, notes about elevation gain, soil composition, etc. The log can be saved to file, and opened. Calculations of cross-section areas are sent to a list box by ADD and the LIST AREA command. Individual area can be removed from the list or cleared before they are added. Selected one method removes or disables text boxes that are not used in the selected method. Common variables such a Center and Roadbed Width remain active. Using the ADD radio button prevents modifying the entries, but the entries still can be cleared. The user can enter cross-sectional areas into the text boxes in the Volume calculator. Data that is not entered into the Log is not saved when the program is closed (Figure 1).

The GUI is set up as a dedicated calculator and note pad for field work. Its purpose is intended for estimating cross-sectional areas and volumes between stations for immediate use. The estimated volume of earth removed or added for the excavations and embankments is rough. Other programs in this set of applications are aimed toward providing a more detailed estimate, if the need requires. Irregular landscape surfaces can involve calculations with a number of variables. The VB code is given on the pages following Figure 1.

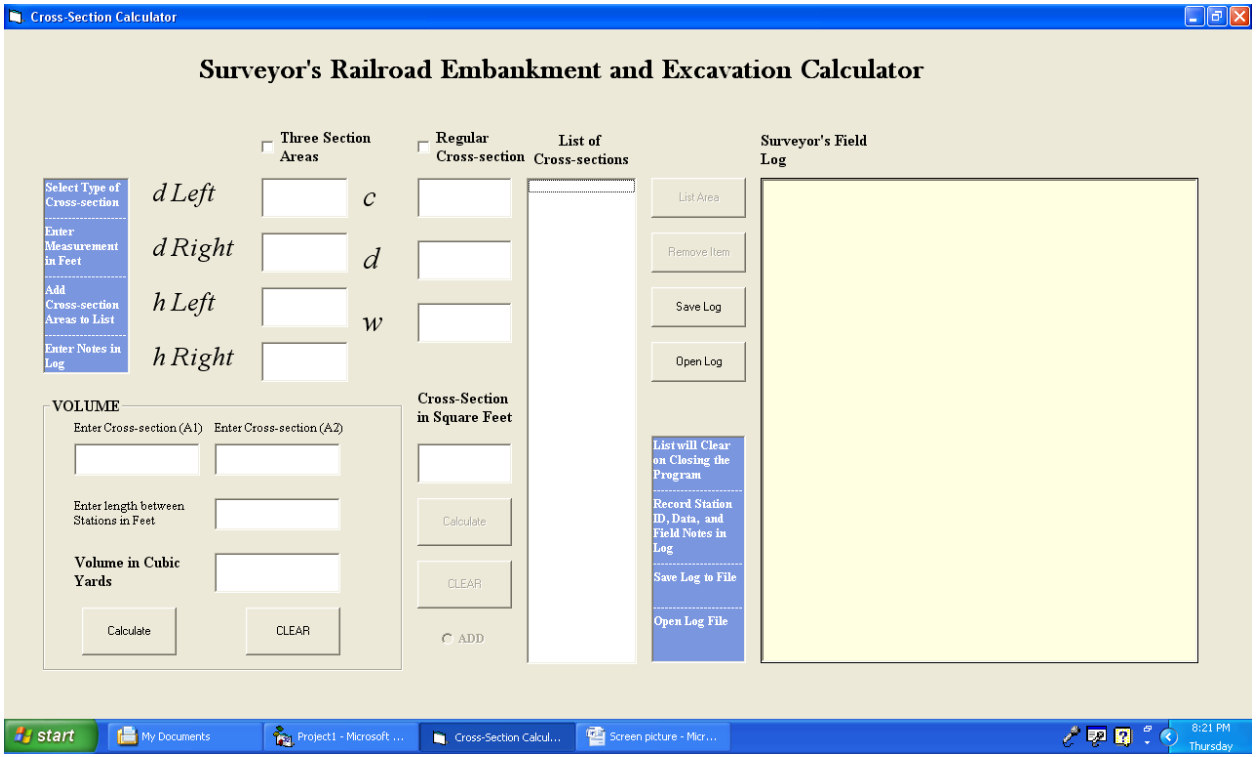

Figure 1. This is GUI for the "Surveyor's Railroad Embankment and Excavation Calculator" program. The VB code for this program is listed below.

```
'James C. Burke
'Railroad Excavations and Embankment Calculator
'24 November 2008
Option Explicit

Private Sub chkReg_Click()
                        'Select Regular Cross-section formula
    cmdCalc.Enabled = True
    cmdClear.Enabled = True
```

```
    chkThree.Enabled = False
    TxtDleft.Enabled = False
    txtDright.Enabled = False
    txtHleft.Enabled = False
    txtHright.Enabled = False
    TxtDleft.Visible = False
    txtDright.Visible = False
    txtHleft.Visible = False
    txtHright.Visible = False
    txtInputC.Enabled = True
    txtInputW.Enabled = True
    txtInputD.Enabled = True
    chkReg.Enabled = False

End Sub

Private Sub chkThree_Click()
                        'Select Three Section formula
    cmdCalc.Enabled = True
    cmdClear.Enabled = True
    txtInputD.Enabled = False
    txtInputD = 1
    txtInputD.Visible = False
    TxtDleft.Enabled = True
    txtDright.Enabled = True
    txtHleft.Enabled = True
    txtHright.Enabled = True
    txtInputC.Enabled = True
    txtInputW.Enabled = True
    chkReg.Enabled = False
    chkThree.Enabled = False

End Sub

Private Sub cmdCalc2_Click()
                        'Volume [V = Length/2 (Cross Sectional Area + Next Cross Section Area)]
```

```
Dim A1 As Double
Dim A2 As Double
Dim L As Double
Dim V As Double
Dim CST As Double

On Error GoTo ErrHandler

    L = txtLength.Text * 0.5
    A1 = txtCS1.Text
    A2 = txtCS2.Text
    CST = A1 + A2
    V = CST * L
    txtVol.Text = V

Exit Sub

ErrHandler:
    MsgBox "Invalid Entry"

End Sub

Private Sub cmdClear_Click()
                        'Clear Cross-section entries
    cmdCalc.Enabled = False
    chkThree.Enabled = True
    chkReg.Enabled = True
    txtInputC = ""
    txtInputW = ""
    TextArea = ""
    TxtDleft = ""
    txtDright = ""
    txtHleft = ""
    txtHright = ""
    chkThree.Value = False
    chkReg.Value = False
```

```
    TxtDleft.Enabled = False
    txtDright.Enabled = False
    txtHleft.Enabled = False
    txtHright.Enabled = False
    TxtDleft.Visible = True
    txtDright.Visible = True
    txtHleft.Visible = True
    txtHright.Visible = True
    txtInputC.Enabled = False
    txtInputD.Enabled = False
    txtInputD = ""
    txtInputD.Visible = True
    txtInputW.Enabled = False

End Sub

Private Sub cmdClear2_Click()
                        'Clear Volume entries
    txtCS1.Text = ""
    txtCS2.Text = ""
    txtLength.Text = ""
    txtVol.Text = ""

End Sub

Private Sub cmdLog_Click()
                        'Enter Cross Sectional Areas in List
    If TextArea = "" Then MsgBox "Error"
    lstArea.AddItem (TextArea.Text)
    txtInputC = ""
    txtInputD = ""
    txtInputW = ""
    TextArea = ""
    TxtDleft = ""
    txtDright = ""
    txtHleft = ""
```

```
    txtHright = ""
                        'Disable entries function
    cmdLog.Enabled = False
    OK.Enabled = False
    chkThree.Enabled = True
    chkReg.Enabled = True
    TxtDleft.Visible = True
    txtDright.Visible = True
    txtHleft.Visible = True
    txtHright.Visible = True
    txtInputD.Visible = True
    TxtDleft.Enabled = False
    txtDright.Enabled = False
    txtHleft.Enabled = False
    txtHright.Enabled = False

End Sub

Private Sub CmdCalc_Click()
                        'Calculates Regular and Three Section Areas
Dim C As Double
Dim D As Double
Dim W As Double
Dim A As Double
Dim Q As Double
Dim S As Double
Dim DLR As Double
Dim HLR As Double
Dim P As Double
Dim R As Double

On Error GoTo ErrHandler
                        'Regular Cross-Section
    C = txtInputC.Text
    D = txtInputD.Text
    W = txtInputW.Text * 0.5
```

```
    Q = D + W
    A = C * Q
    TextArea.Text = A

If chkReg.Value = False Then
                                'Three Section
    C = txtInputC.Text * 0.5
    W = txtInputW.Text * 0.25
    DLR = TxtDleft.Text + txtDright.Text
    HLR = txtHleft.Text + txtHright

    P = W * HLR
    R = C * DLR

    A = P + R
    TextArea.Text = A

    End If

OK.Enabled = True
OK.Value = False
chkReg.Value = False
chkThree.Value = False
txtInputC.Enabled = False
txtInputD.Enabled = False
txtInputW.Enabled = False

Exit Sub

ErrHandler:
    MsgBox "Invalid Entry"

End Sub
```

```
Private Sub CmdOpen_Click()
                        'Opens Log File
    Dim strOpenFile As String
    On Error GoTo ErrHandler
    CommonDialog1.CancelError = True
    CommonDialog1.Flags = cdlOFNFileMustExist + cdlOFNHideReadOnly
    CommonDialog1.Filter = "All Files (*.*)|*.*|Rich Text Files" & "(*.rtf)|*.rtf|Text Files(*.txt)*.txt"
    CommonDialog1.FilterIndex = 2
    CommonDialog1.ShowOpen
    strOpenFile = CommonDialog1.FileName
    RichTextBox1.LoadFile strOpenFile

ErrHandler:

Exit Sub

End Sub

Private Sub cmdRemove_Click()

On Error GoTo ErrHandler

                        'Removes List Entries
If lstArea.ListIndex <> -1 Then
Call lstArea.RemoveItem(lstArea.ListIndex)

End If

ErrHandler:
    MsgBox "Removed Item"

End Sub

Private Sub cmdSave_Click()
                        'Saves Log as file
Dim strNewFile As String
```

```
On Error GoTo ErrHandler
    CommonDialog1.CancelError = True
    CommonDialog1.Flags = cdlOFNHideReadOnly + cdlOFNOverwritePrompt
    CommonDialog1.Filter = "All Files (*.*)|*.*|Rich Text Files" & "(*.rtf)|*.rtf|Text Files(*.txt)*.txt"
    CommonDialog1.FilterIndex = 2
    CommonDialog1.ShowSave
    strNewFile = CommonDialog1.FileName
    RichTextBox1.SaveFile strNewFile

ErrHandler:

Exit Sub

End Sub

Private Sub OK_Click()
                        'Approve Calculation of Area and sends answer to be added to List
cmdLog.Enabled = True
chkThree.Enabled = False
chkReg.Value = 0
OK.Enabled = False
cmdRemove.Enabled = True
End Sub
```

Project 2: Irregular Road Cross-sections

As stated above, the formula for Irregular Road Cross-sections can be expressed as $H_2\left(+\frac{w}{2}-D_1\right)+H_1(+D_2-0)+C(+D_1+d_1)+h_1(0+d_2)+h_2(-d_1+\frac{w}{2})$. In this program the center measurement, road width, heights and distances can be entered into the text boxes on the GUI (Figure 2).

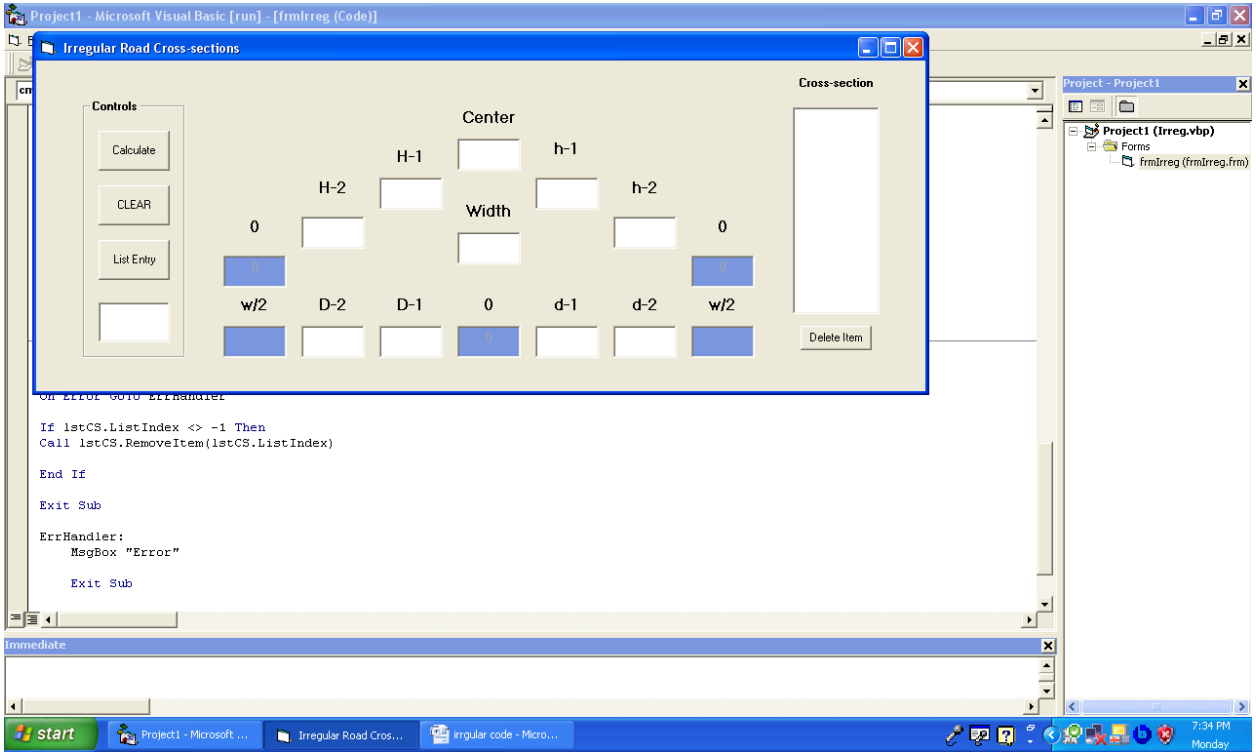

Figure 2. The GUI for the Irregular Road Cross-sections calculator is organized for entering height and distance measurements. The boxes for "0" and "w/2" in blue are disabled.

The elements in the blue text boxes include “0” and half the width of the road bed. These boxes are disabled. The calculations can be saved to the list box to the right of the form. This program is intended to be a function that can be brought up in the main program.

The VB code for this function is as follows.

```
'James C. Burke
'Irregular Cross-sections
'28 November 2008

Option Explicit

Private Sub cmdCalc1_Click()

Dim C As Double
Dim d1 As Double
Dim d2 As Double
Dim h1 As Double
Dim h2 As Double
Dim W As Double
Dim DD1 As Double
Dim DD2 As Double
Dim HH1 As Double
Dim HH2 As Double
Dim WL As Double
Dim WR As Double
Dim CS As Double
Dim B As Double
Dim Q As Double
Dim R As Double
Dim T As Double
Dim U As Double
Dim V As Double

 On Error GoTo ErrHandler
```

```
    W = txtwidth.Text
    txtwLeft.Text = W * 0.5
    txtwRight.Text = W * 0.5
    C = txtCenter.Text
    d1 = txtd1right.Text
    d2 = txtd2right.Text
    DD1 = txtDD1left.Text
    DD2 = txtDD2left.Text
    h1 = txth1right.Text
    h2 = txth2right.Text
    HH1 = txtHH1left.Text
    HH2 = txtHH2left.Text
    WL = txtwLeft.Text
    WR = txtwRight.Text

        B = HH2 * (txtwLeft.Text - DD1)
        Q = DD2 * HH1
        R = C * (DD1 + d1)
        T = d2 * h1
        U = h2 * (txtwRight.Text - d1)

        CS = B + Q + R + T + U

  txtansr.Text = CS * 0.5

    Exit Sub

ErrHandler:
    MsgBox "Error"

    Exit Sub

cmdAdd.Enabled = True

End Sub
Private Sub cmdAdd_Click()
```

```
lstCS.AddItem (txtansr.Text)
txtansr.Text = ""

cmdAdd.Enabled = False

End Sub

Private Sub CmdClear1_Click()

txtCenter.Text = ""
txtd1right.Text = ""
txtd2right.Text = ""
txtDD1left.Text = ""
txtDD2left.Text = ""
txth1right.Text = ""
txth2right.Text = ""
txtHH1left.Text = ""
txtHH2left.Text = ""
txtwidth.Text = ""
txtwLeft.Text = ""
txtwRight.Text = ""

End Sub

Private Sub cmdCearitem_Click()

On Error GoTo ErrHandler

If lstCS.ListIndex <> -1 Then
Call lstCS.RemoveItem(lstCS.ListIndex)

End If

Exit Sub
```

```
ErrHandler:
    MsgBox "Error"

    Exit Sub

cmdAdd.Enabled = True

End Sub
```

Project 3: Simpson's One-third Rule

Simpson's One-third Rule is used for calculating the area bounded by the arc of a parabola where d is a half the distance from the center $\boldsymbol{h_2}$ with h_1 and h_3 as the two end measurements. The formula is expressed as $A_{1,2} = \frac{h_1+h_3}{2}2d + \left(h_2 - \frac{h_1+h_3}{2}\right)2d \times \frac{2}{3} = \frac{d}{3}(h_1 + 4h_2 + h_3)$. Measurement h_2 is equidistant from h_1 and h_3.

The GUI for the program is user friendly. The heights and distance measurements are all that is required. This program is designed as a function of the general program (Figure 3).

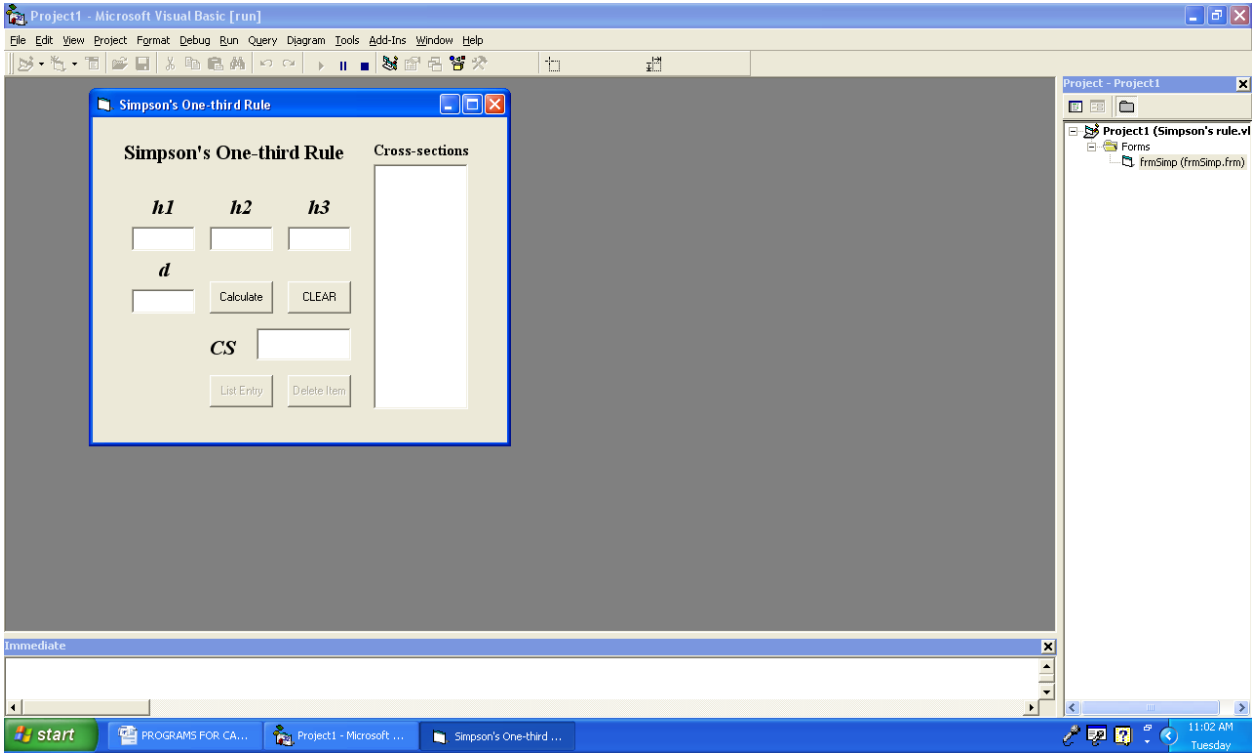

Figure 3. This function calculates cross-sectional area according to Simpson's One-third Rule.

Depending on the concavity or convexity of the curved surface the results might be greater or less than results obtained by other methods. [A variant of Simpson's One-third Rule is the Prismodial Formula for calculating volume, $V = \frac{l}{6}(A_1 + 4A_m + A_2)$. In this formula the middle cross-section is determined by averaging the dimensions of the end sections.] The code for the function appears below.

```
'James C. Burke
'Simpson 's One-third Rule
'2 December 2008
Option Explicit
Private Sub cmdCalc_Click()

Dim h1 As Long
Dim h2 As Long
Dim h3 As Long
Dim d As Long
Dim Hsum As Long

On Error GoTo ErrHandler

   h1 = txth1.Text
   h2 = txth2.Text * 4
   h3 = txth3.Text
   d = txtd.Text / 3

   Hsum = h1 + h2 + h3
   txtCS.Text = d * Hsum


cmdAdd.Enabled = True

   Exit Sub
```

```
ErrHandler:
   MsgBox "Error"

   Exit Sub

End Sub
Private Sub cmdAdd_Click()

lstCS.AddItem (txtCS.Text)
txtCS.Text = ""

cmdAdd.Enabled = False
cmdDelete.Enabled = True

End Sub
Private Sub cmdClear_Click()

txth1.Text = ""
txth2.Text = ""
txth3.Text = ""
txtd.Text = ""
txtCS.Text = ""

cmdDelete.Enabled = False

End Sub
Private Sub cmdDelete_Click()

On Error GoTo ErrHandler

If lstCS.ListIndex <> -1 Then
Call lstCS.RemoveItem(lstCS.ListIndex)

End If

Exit Sub
```

```
ErrHandler:
    MsgBox "Error"

Exit Sub

cmdAdd.Enabled = True

End Sub
```

Project 4: The Prismoidal Formula

The Prismoidal Formula is another application of Simpson's One-third Rule. The program associated with this procedure calculates the middle dimensions from the measurements from the dimensions of the station before and after. It calculates the areas in square feet for all three, and provides the volume of the entire section in cubic yards (Figure 4).

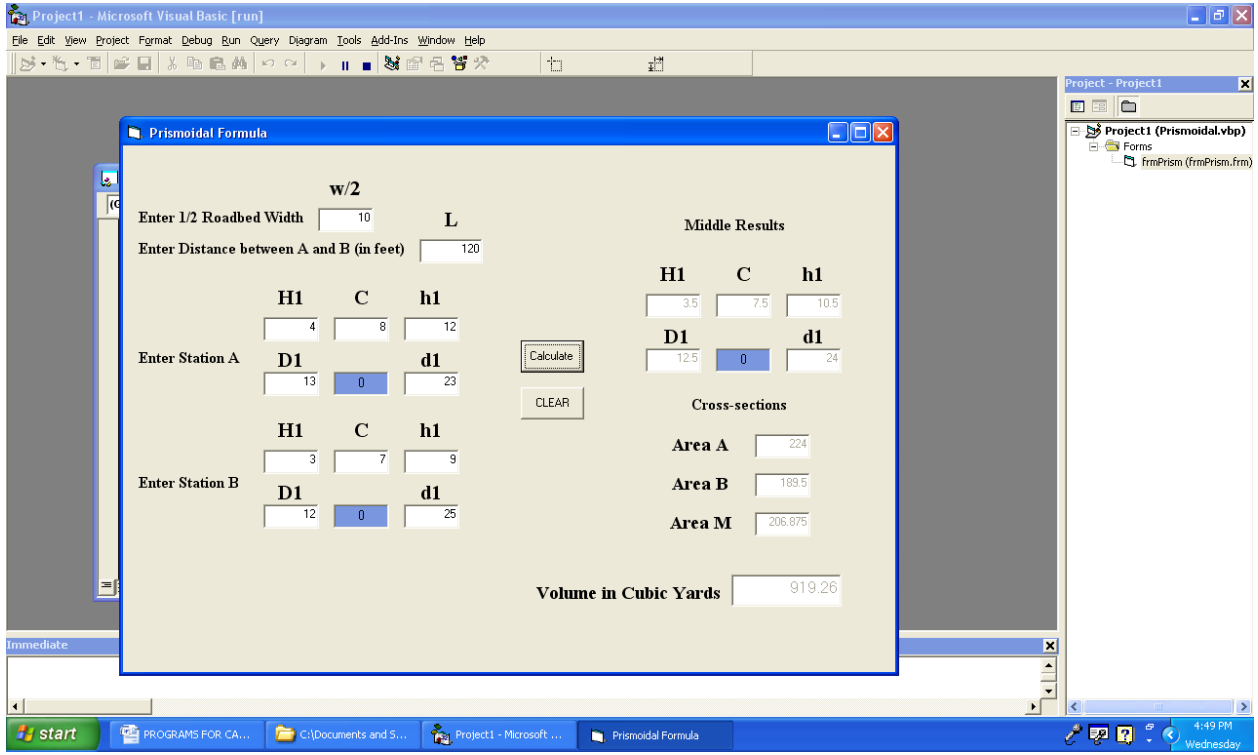

Figure 4. This figure shows the GUI for the Prismoidal Formula application. The user enters the measures for stations A and B as well as the length between them and one-half the width of the roadbed. After clicking the command "Calculate," the program generates the dimensions of the middle cross-section, the areas of all three cross-sections, and the volume of the entire section.

The VB code for the procedure is given below.

```
Option Explicit

Private Sub cmdMid_Click()

Dim HH1 As Double
Dim HH2 As Double
Dim HH3 As Double
Dim C1 As Double
Dim C2 As Double
Dim C3 As Double
Dim h1 As Double
Dim h2 As Double
Dim h3 As Double
Dim DD1 As Double
Dim DD2 As Double
Dim DD3 As Double
Dim d1 As Double
Dim d2 As Double
Dim d3 As Double
Dim A1 As Double
Dim A2 As Double
Dim A3 As Double
Dim W As Double
Dim D As Double
Dim E As Double
Dim V As Double

On Error GoTo ErrHandler

   HH1 = txtHH1.Text
```

```
    HH2 = txtHH2.Text
    HH3 = (HH1 + HH2) / 2

txtHH3.Text = HH3

    C1 = txtc1.Text
    C2 = txtc2.Text
    C3 = (C1 + C2) / 2

txtc3.Text = C3

    h1 = txth1.Text
    h2 = txth2.Text
    h3 = (h1 + h2) / 2

txth3.Text = h3

    DD1 = txtDD1.Text
    DD2 = txtDD2.Text
    DD3 = (DD1 + DD2) / 2

txtDD3.Text = DD3

    d1 = txtd1.Text
    d2 = txtd2.Text
    d3 = (d1 + d2) / 2

txtd3.Text = d3

    A1 = HH1 * txtw.Text + C1 * (DD1 + d1) + h1 * txtw.Text
    A2 = HH2 * txtw.Text + C2 * (DD2 + d2) + h2 * txtw.Text
    A3 = HH3 * txtw.Text + C3 * (DD3 + d3) + h3 * txtw.Text

txtA1.Text = A1 / 2
txtA2.Text = A2 / 2
txtA3.Text = A3 / 2
```

```
    D = txtdist.Text / 6
    E = txtA1.Text + 4 * txtA3.Text + txtA2.Text
    V = D * E / 27

txtVol.Text = Round(V, 2)

    Exit Sub

ErrHandler:
    MsgBox "Error"

End Sub
Private Sub cmdClear_Click()

txtw.Text = ""
txtdist.Text = ""
txtc1.Text = ""
txtc2.Text = ""
txtc3.Text = ""
txtHH1.Text = ""
txtHH2.Text = ""
txtHH3.Text = ""
txth1.Text = ""
txth2.Text = ""
txth3.Text = ""
txtDD1.Text = ""
txtDD2.Text = ""
txtDD3.Text = ""
txtd1.Text = ""
txtd2.Text = ""
txtd3.Text = ""
txtA1.Text = ""
txtA2.Text = ""
txtA3.Text = ""
txtVol.Text = ""
```

```
End Sub
```

The Full Program

The complete program includes all the component applications that can be called from the three commands above the Rich Text Box of the Surveyor's Field Log (Figure 5).

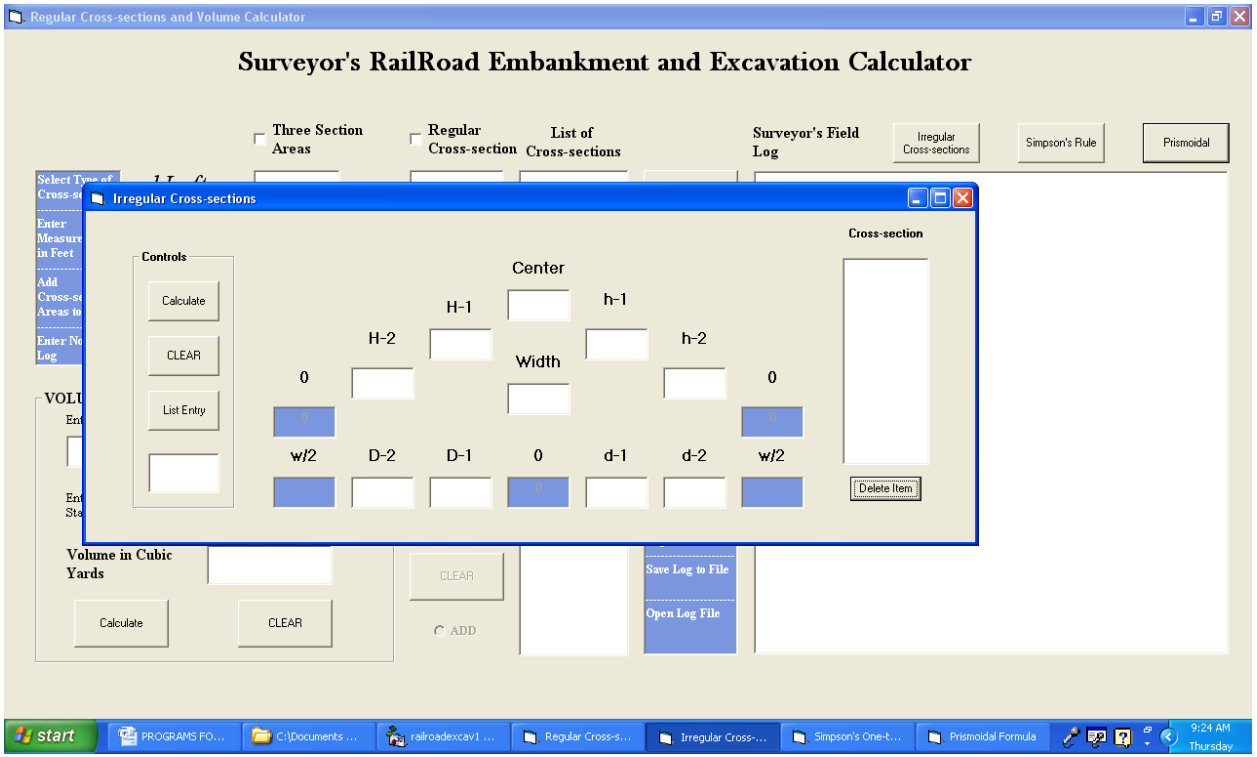

Figure 5. The entire program contains all the individual tools for making field calculations for railroad embankments and excavations. Each procedure is accessible from the main GUI.